CLAIMS

Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is:

Sub a1

1. A method for identifying to a user, the differences between elements of two hierarchically structured files, comprising the steps of:

comparing the elements of a base file to the elements of a modified file;

providing to the user a tree structure, said tree structure combining the elements of said base and said modified files; and

highlighting the differences between said elements of said base and said modified files.

2. The method of claim 1 further comprising the step of allowing the user to resolve said differences between elements, thereby creating a merged file containing elements from said base file and elements from said modified file.

3. The method of claim 2 which includes indicating to the user differences between elements by one of the identifiers: new, changed or removed.

4. The method of claim 3 which includes, for an element identified as new, providing the user with the following options:

a) do not use the new element, whereby the new element is not incorporated into said merged file; and

b) use the new element, whereby the new element and children thereof, if any, are incorporated into said merged file.

5. The method of claim 3 which includes, for an element identified as changed, providing the user with the following options:

a) use old, where conflict, whereby for the merged file the changed element is taken from the base file together with unresolved children thereof, if any; and

b) use new, where conflict, whereby for the merged file the changed element is taken from the modified file together with unresolved children thereof, if any.

6. The method of claims 3, which includes, for an element identified as removed, providing the user with the following options:

a) do not delete, whereby the merged file has the element as it exists in the base file; and

b) delete from the base file, whereby the merged file does not have the element that was deleted from the base file.

7. The method of claim 1 wherein the step of providing to user a tree structure comprises visually displaying the tree structure.

8. The method of claim 7 wherein visually displaying the tree structure comprises displaying to the user a screen containing three panes, the first pane

displaying said tree structure, the second pane displaying an element of said base file, and the third pane displaying an element of the modified file.

9. The method of claim 8 which includes, when the user selects an element of the tree structure displayed in the first pane, displaying the source code for the selected element:

a) in the second pane if the selected element exists in the base file; and

b) in the third pane if the selected element exists in the modified file.

10. The method of claim 1 wherein the step of comparing uses an ID attribute of the elements of the base file and the modified file being compared.

11. The method of claim 1 wherein the step of comparing uses a name attribute of the elements of the base file and the modified file being compared.

12. The method of claim 1 wherein said hierarchically structured files are XML (eXtensible markup language) files and wherein the step of comparing uses:

if provided by the elements of the base and modified files being compared, an attribute of type ID;

if an attribute of type ID is not provided by the elements of the base and modified files being compared, a <Uuid> tag if provided by the elements of the base and modified files being compared;

if an attribute of type ID and a <Uuid> tag is not provided by the elements of the base and modified files being compared, a name attribute if provided by the elements of the base and modified files being compared; and

if an attribute of type ID, a <Uuid> tag and a name attribute is not provided by the elements of the base and modified files being compared, a concatenation of a tag of the element and a value of the element.

13. The method of claim 1 wherein said hierarchically structured files are XML (eXtensible Markup Language) files.

14. A method for visually identifying to a user, the differences between elements of a hierarchical base data structure and a hierarchical modified data structure, comprising the steps of:

comparing the elements of said base data structure to the elements of said modified data structure;

displaying to the user a tree structure, said tree structure combining the elements of said base and modified data structures; and

highlighting the differences between said elements of said base and modified data structures.

15. A program storage device readable by a data processing system, tangibly embodying a program of instructions, executable by said data processing system to perform the method steps of claim 1.

16. A system for identifying to a user, the differences between elements of two hierarchically structured files, comprising:

means for comparing the elements of a base file to the elements of a modified file;

means for providing to the user a tree structure, said tree structure combining the elements of said base and said modified files; and

means for highlighting the differences between said elements of said base and said modified files.

17. The system of claim 16 further comprising means for allowing the user to resolve said differences between elements, thereby creating a merged file containing elements from said base file and elements from said modified file.

18. The system of claim 17 which includes means for indicating to the user differences between elements by one of the identifiers: new, changed or removed.

19. The system of claim 18 which includes, for an element identified as new, providing the user with the following options:

a) do not use the new element, whereby the new element is not incorporated into said merged file; and

b) use the new element, whereby the new element and children thereof, if any, are incorporated into said merged file.

20. The system of claim 18 which includes, for an element identified as changed, means for providing the user with the following options:

a) use old, where conflict, whereby for the merged file the changed element is taken from the base file together with unresolved children thereof, if any; and

b) use new, where conflict, whereby for the merged file the changed element is taken from the modified file together with unresolved children thereof, if any.

21. The system of claim 18, which includes, for an element identified as removed, means for providing the user with the following options:

a) do not delete, whereby the merged file has the element as it exists in the base file; and

b) delete from the base file, whereby the merged file does not have the element that was deleted from the base file.

22. The system of claim 16 wherein the means for providing to user a tree structure comprises means for visually displaying the tree structure.

23. The system of claim 22 wherein the means for visually displaying the tree structure comprises means for displaying to the user a screen containing three

panes, the first pane displaying said tree structure, the second pane displaying an element of said base file, and the third pane displaying an element of the modified file.

24. The system of claim 23 which includes, when the user selects an element of the tree structure displayed in the first pane, means for displaying the source code for the selected element:

a) in the second pane if the selected element exists in the base file; and

b) in the third pane if the selected element exists in the modified file.

25. The system of claim 16 wherein the means for comparing uses an ID attribute of the elements of the base file and the modified file being compared.

26. The system of claim 16 wherein the means for comparing uses a name attribute of the elements of the base file and the modified file being compared.

27. The system of claim 16 wherein said hierarchically structured files are XML (eXtensible markup language) files and wherein the means for comparing uses:

if provided by the elements of the base and modified files being compared, an attribute of type ID;

if an attribute of type ID is not provided by the elements of the base and modified files being compared, a <Uuid> tag if provided by the elements of the base and modified files being compared;

if an attribute of type ID and a <Uuid> tag is not provided by the elements of the base and modified files being compared, a name attribute if provided by the elements of the base and modified files being compared; and

if an attribute of type ID, a <Uuid> tag and a name attribute is not provided by the elements of the base and modified files being compared, a concatenation of a tag of the element and a value of the element.

28. The system of claim 16 wherein said hierarchically structured files are XML (eXtensible Markup Language) files.

29. A system for determining the differences between two hierarchically structured files comprising:

a parser to parse the files and produce a parse tree output for each file; and

a comparison module to compare the parse trees output from the parser and to create a merged tree from the parse tree outputs.

30. The system of claim 29, further comprising a tree view module to display the merged tree.

31. A hierarchical data structure for use by a computer system and stored on a computer-readable storage medium, said structure comprising:

a plurality of nodes;

each of said nodes corresponding to a hierarchical element contained within a base file or a modified file, said files stored within said computer system; and

each of said nodes having an indicator if said node is new, changed or removed when comparing the nodes of said base file to said modified file.